DELHI PUBLIC SCHOOL BANGALORE SOUTH

LAB ACTIVITY – 3

Topic: File Handling

TEXT FILES

Instructions:

a) Create a sample text file for each of the programs below. The file names shall represent the question number: textfile1, textfile2,..etc
b) Use exception handling techniques to handle invalid input as well as file read/write operations.
c) Use appropriate built in functions to perform string operations.
d) Create user defined functions for each progam.

1. Read a text file that contains information as below. Extract all dates in all formats. Convert them to yyyy/mm/dd format. Store them back in the file. (modification)

   Jane Smith is the first student and her student number is D1234567, and she joined DIT on 01/09/2013. John Smith is the second student and his student number is D1234568, and he joined DIT on the 10th September 2013. Jo Smith is the third student and her student number is D1234569, and she joined DIT on 01/09/2014. Joe Smith is another student and his number is <NOT SURE> ,and he joined DIT on 1st of Sept. 2014.

2. Read a text file containing 'n' lines. Create another text file deleting the words "a", "the","an" and replacing each one of them with a blank space. (deleting)
3. Given a list of names of students in a class - write a program to store the names in a file on disk, (i) display the nth name in the list(n is data to be read), (ii) display all names starting with S. (iii) Add more names to the file. (append)
4. The file Q4.TXT, contains 'n' lines. Read the file and reverse words starting with a vowel and store it back on the file. Display the edited file.
5. The file Q5.txt contains 'n' lines. Read the file and display:
   a) Number of 4 letter words.
   b) Count of digits and symbols

BINARY FILES

Instructions:

a) Include a try except block to handle : EOFError , pickle.PicklingError, pickle.UnpicklingError
b) Close the file after required set of operations.
c) Use pickle module for the operations.

1. Create a program to store ‘n’ aircraft incident records. Perform the following tasks on the same.

a) Make_Data(op=0) – The dictionary format to store data of an aircraft incident is given below:

Key	Value type
Report_id	Str
Date	datetime.date or any sequence(str,tuple)
Airport	Str
aircraft_id	Str
aircraft_type	Str
pilot_total_hours	Int (>0)
Midair	Bool

Accept the number of records from the user and write each dictionary to the file: BINARY1.DAT, in ‘wb’ mode (creating) or ‘ab’ mode (if op is 1, adding more records).

b) Get_Data(): This function reads each dictionary from BINARY.DAT. It accepts aircraft_id from the user and displays the matching record, if found, else displays ‘NO SUCH RECORD’.

c) Update_data():
This function accepts **report_id** from the user and modifies it, if a matching record is found, else displays ‘NO SUCH RECORD’.
(i) Use tell(), and modify the same file, if the value has the same size as original.
(ii) If the current value is larger than original, use a new file to copy old and modified records.
Note: use sys.getsizeof() to determine the size of data.

Main segment:

1. Call Make_Data() to create the file.
2. Provide the options 1/2/3 for displaying a required record, updating or appending data, respectively. For option 3: set op=1, in the function call.

2. Create a program to store ‘n’ dictionaries as given below. Perform the following tasks on the same.

Key	Value type
City_Name	String
Population	Int
Hospitals	Int
Schools	Int
density	Int

a) Make_Data(op=0) Accept the number of records from the user and write each dictionary to the file: BINARY2.DAT, in ‘wb’ mode (creating) or ‘ab’ mode (if op is 1, adding more records).
b) Get_Data(): Accept a city name and display corresponding record. Display ‘NOT FOUND’ , if the name does not exist.
c) Update_Data(): Replace City_Name with ‘Unknown’, if density is in the range: 500 to 1000, both inclusive.
d) Update_Rec(): Accept city name from the user, if found in the file, modify the same. Use tell() and seek to modify the file. (Do not create a new file.)

3. Create a program to store ‘n’ dictionaries as given below. Perform the following tasks on the same.

Key	Value type
Court_est	String
Police_Station	String
FIR	Int
Year	Int (between 1990 and current year)
Status	Pending/Disposed

a) Make_Data(op=0) Accept the number of records from the user and write each dictionary to the file: BINARY3.DAT, in ‘wb’ mode (creating) or ‘ab’ mode (if op is 1, adding more records).
b) Get_Data(): Accept a FIR and display corresponding status. Display ‘NOT FOUND’ , if the name does not exist.
c) Update_Data(): Change the status to disposed , for all years < 2005. Display edited file.

CSV FILES

Note:
a) Each row shown in the table must be used as a list. Input the values from the user for each list.
b) Display all results in a tabular form with respective headers.

1. Create a CSV file to store the following data.

Amount	Average	People
279000		15
396800		36
563000		44
150000		19

716200		58

Compute Average as: Amount/People and store the data.
Perform the following operations:

a) Read the file. Print the data in a tabular form.
b) Display first and last columns.
c) Remove rows with people count less than 20.

2. Create a CSV file to store the following data, using the delimiter: '|'. Perform the following operations:

Name	Score	Attempts	Qualify
Anastasia	12.5	1	yes
Dima	9	3	no
Katherine	16.5	2	yes
James	0	3	no
Emily	9	2	no
Michael	20	3	yes
Matthew	14.5	1	yes
Laura	0	1	no
Kevin	8	2	no
Jonas	19	1	yes

a) Read the file. Compute mean score: sum of terms/ number of terms
b) Replace all 0's with None.
c) Display the transpose of data.

3. Create a CSV file to store the following data. Perform the following operations:

First Name	Last Name	Date_Of_Birth	Age
Alberto	Franco	17/05/2002	
Gino	Mcneill	16/02/1999	
Ryan	Parkes	25/09/1998	
Eesha	Hinton	11/5/2002	
Syed Ryan	Wharton Parkes	15/09/1997 25/09/1998	

Compute age and then store the data.

a) Read the file. Display all last names starting with 'M' or 'P'.
b) Remove duplicate rows.
c) Create a new file which has the data in sorted order of age.